REMARKS

In the final Office Action, the Examiner: (i) rejects claims 1, 3, 13, 15, 25 and 26 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,161,136 (hereinafter "Hyndman"); and (ii) rejects claims 2, 4-12, 14 and 16-24 under 35 U.S.C. §103(a) as being unpatentable over Hyndman in view of U.S. Patent No. 5,768,510 (hereinafter "Gish").

In this response, Applicants respectfully traverse the §102 and §103 rejection of claims 1-26.

Regarding the §102(e) rejection of claims 1, 3, 13, 15, 25 and 26, the final Office Action contends that Hyndman discloses all of the claim limitations recited in the subject claims. Applicants respectfully assert that Hyndman fails to teach or suggest all of the limitations in claims 1, 3, 13, 15, 25 and 26, for at least the reasons presented below.

It is well-established law that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). Applicants assert that the rejection based on Hyndman does not meet this basic legal requirement, as will be explained below.

The claimed invention, as recited in independent claim 1, provides a method for use in a client/server system of reducing interactions between a client and server in association with an application being accessed by the client at the server. The method comprises the steps of: configuring the server to store a model associated with the application and to execute view-generating and controller logic associated with the application; and configuring the client to store at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application; wherein one or more portions of the application are performed at the client without the client having to interact with the server, and further wherein the client and the server both locally maintain at least a portion of the model and execute the view-generating and controller logic associated therewith. Independent claims 13, 25 and 26 recite similar limitations.

As explained on page 3, lines 22-27, of the present specification: "[t]he invention addresses performance by employing a dual-MVC approach, in which a subset of the application's Model-

View-Controller reside on the client, and the full Model-View-Controller and View-Generating-Logic reside on the server, thereby reducing the number of required server interactions." FIG. 3 of the present application illustrates such an inventive dual-MVC approach.

Thus, the claimed invention recites that the server "store[s] a model associated with the application and maintain[s] view-generating and controller logic associated with the application," and the client "store[s] at least a subset of the model associated with the application and maintain[s] at least a subset of the view-generating and controller logic associated with the application."

While Hyndman discloses a multilevel model-view-controller (MMVC), Hyndman does not disclose the dual-MVC approach of the claimed invention. That is, among other deficiencies, Hyndman does not disclose having a server to store and maintain a model associated with the application and to execute view-generating and controller logic associated with the application, and having a client to store and maintain at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application.

The Office Action relies on Hyndman at column 2, lines 40-49, which states the following with emphasis supplied:

> According to a further aspect of the invention, there is provided a method of structuring functionality of a user interface for a network management system of a communication network into a multilayered model-view-controller pattern, comprising separating the user interface into a server and a client layer integrating the state information on the server in a high-level model, integrating the state information on the client layer as a high-level view, and integrating a data marshalling mechanism provided by the network management system as a high-level controller.

The Office Action also relies on Hyndman at column 3 line 59-64, which states the following with emphasis supplied:

> At the high level, the two parts of the user interface according to the invention, namely the UIC and UIS, follow this MVC pattern. The UIS state information forms the model, denoted with M, and the data marshalling mechanism is the controller, denoted with C, and the view is the UIC portion of the interface, denoted with V.

> FIG. 1B illustrates the next level of the MMVC architecture according to the invention. For example, MVC-1, comprising view V-1, model M-1 and controller C-1, is the view V-2 for the next level MVC-2. MVC-2, comprising view V-2, model M-2 and controller C-2, is the view V2 for the next level MVC-3. This continues until the finest granularity for any given system.
>
> If the MMVC pattern is applied to the structure shown in FIG. 1B, at a high level, the view V-2 of the MVC-2 is the UIC state information, the model M-2 is the UIS and the controller C-2 is the data marshalling mechanism for the various applications provided in the MN.

The relied-upon portions of Hyndman do not teach or suggest having a server to store and maintain a model associated with the application and to execute view-generating and controller logic associated with the application, and having a client to store and maintain at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application.

In response to Applicants' arguments, the Examiner refers to FIG. 1B and column 4, lines 1-25 of Hyndman as disclosing a server to store and maintain a model associated with the application and to execute view-generating and controller logic associated with the application, and disclosing a client to store and maintain at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application. In particular, the Examiner points to MVC-2 as teaching or suggesting the server layer MVC, and MVC-1 as teaching or suggesting the client layer MVC on page 2, fifth paragraph of the Final Office Action.

The Examiner appears to be suggesting that MVC-2 is a server storing and maintaining a model M-2 and executing view-generating with V-2 and controller logic with C-2, where M-2, V-2 and C-2 are associated with the application, and MVC-1 is a client storing and maintaining at least a subset of the model associated with the application (M-1) and executing at least a subset of the view generating with V-1 and controller logic with C-1 associated with the application. However, Hyndman at column 4, lines 6-8, states that "the view V-2 of MVC-2 is the UIC state information, the model M-2 is the UIS and the controller C-2 is the data marshalling mechanism for the various applications provided in the MN." Therefore, V-2, as the UIC state information, does not teach or suggest the claimed view-generating logic that is associated with the application and executed by the server, and C-2, as the marshalling mechanism for the various applications provided in the MN, does not teach or suggest the claimed controller logic associated with the application and executed by the server. Furthermore, the relied-upon portion of Hyndman does not teach or suggest of a client storing and maintaining at least a subset of the model associated with the application and executing at least a subset of the view-generating and controller logic associated with the application. For example, in the relied-upon portion of Hyndman, the only reference to the client side controller logic, C-1, states that, "UIC comprises one or more mini-controllers C-1 that interact both with the model cache M-1 and the controllers C-2 on the server, to ensure that both models are kept in synchronization" (see Hyndman at column 4, lines 16-19). The relied-upon portion of Hyndman does not disclose the limitation of a client executing at least a subset of the controller logic associated with the application.

Thus, the architecture of Hyndman does not teach or suggest having a server to store and maintain a model associated with the application and to execute view-generating and controller logic associated with the application, and having a client to store and maintain at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application.

Accordingly, Applicants assert that independent claims 1, 13, 25 and 26, as well as the claims which depend therefrom, are patentable over Hyndman and therefore allowable. Such dependent claims also recite patentable subject matter in their own right.

Regarding the §103(a) rejections to claims 2, 4-12, 14 and 16-24, Applicants respectfully assert that such dependent claims are patentable over the Hyndman/Gish combination for at least the reasons given above with respect to independent claims 1 and 13. Gish fails to remedy the deficiencies of Hyndman. However, Applicants also assert that such dependent claims also recite patentable subject matter in their own right.

In view of the above, Applicants believe that claims 1-26 are in condition for allowance, and respectfully request withdrawal of the §102 and §103 rejections.

Respectfully submitted,

[illegible]

Date: April 24, 2007

William E. Lewis
Attorney for Applicant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946